

[illegible]

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Systems and Methods for Switching Internet Contexts Without Process Shutdown

Inventor(s):

Ahsan S. Kabir

Erik Snapper

Darren Mitchell

Rajeev Dujari

ATTORNEY'S DOCKET NO. MS1-569US

1 **TECHNICAL FIELD**

2 This invention relates to managing network connections and, more
3 particularly, to switching Internet contexts without requiring process shutdown.
4

5 **BACKGROUND**

6 When a user browses the Internet using a browser on a computer, the user
7 builds up an Internet context for her browsing activity, e.g., through cookies,
8 history and personalized data. As the popularity of Internet browsing has grown,
9 it is not uncommon to have more than one person using a single computer. When
10 different people use the same computer for Internet browsing, Internet contexts
11 can become confused. As a result, one user may no longer know which websites
12 he has visited, and users' preferences for the same website may conflict.

13 To accommodate multi-user arrangements, operating systems support
14 "profiles" for different users, each profile defining a particular work environment
15 configured for a particular user. Each user of a computer can maintain his or her
16 own profile, which accommodates the particular user. However, switching from
17 one Internet context to another requires a system-wide process shutdown, meaning
18 that a current user must end any tasks that are in progress and log off before a new
19 user can log in. In addition, most families do not go to the trouble to enable
20 profiles on their home computers and, therefore, they cannot switch contexts at all.
21
22
23
24
25

SUMMARY

The implementations described herein allow web browsers and other Internet client applications to provide separate contexts for different users, or identities without requiring a process shutdown. This is accomplished by creating an Internet context management component that can select new directory locations for storing non-content state and by tagging content to indicate whether or not the data is personalized to a specific user.

In one implementation, an Internet management object maintains a set of objects called containers, one for each class of Internet state. Each container is located in its own directory on the file system of a computer. This set of containers is called a cache. The information in these containers forms an Internet context that is associated with an identity, each computer user having a unique identity.

When a client wants to change Internet contexts, it supplies a globally unique identifier (guid) that denotes a specific identity to the Internet management object. This initiates the process by which the Internet management object will switch Internet contexts.

The Internet management object shuts down the cache for a current user to prevent any operations from utilizing the cache any further. To simplify naming directories and content tagging, the Internet management object associates each guid with an ordinal according to one described implementation. The Internet management object attempts to create and use a set of containers associated with the new guid using this ordinal. If this is unsuccessful, the Internet management object will restart the cache and continue to use the current Internet context.

1 Otherwise, the Internet management object flushes any authentication credentials,
2 purges session cookies and resets the session start time. In other words, the
3 Internet browser or client application will behave as if it has been closed and
4 restarted.

5 As the Internet context is built with the new identity, cookies, history and
6 other non-content for the new context are placed in a directory associated with the
7 ordinal that maps to the identity's guid. However, content cache is handled
8 differently.

9 To avoid duplicating Internet content across multiple directories (and thus
10 using large amounts of disk space) content is shared across all identities. But there
11 are cases in which an item is meant for only one identity and there are cases
12 wherein the same universal resource locator (URL) will map to different content
13 for different identities. For example, if a web page is a secure page that only some
14 users can access, or if a page uses cookies, etc.

15 When content is shared, it is stored in a common directory that is used by
16 each identity. This directory contains an index file that has multiple records
17 indicating the location and other information for one or more web pages. When
18 information from a web page is stored, the URL for the web page is hashed and
19 the hash value is used to look up an index record. The index record reveals
20 information regarding this web page data, including a file name where the
21 information is stored.

22 When content is user-specific, information indicating that the content is not
23 to be shared is provided to the Internet management object. When the Internet
24 management object detects that this indication has been provided, it performs the
25 look up by hashing a combination of the URL and the ordinal associated with the

1 appropriate identity. This locates a unique user-specific index record similar to
2 that used for shared-content, except that the information is specific for the user,
3 which only the proper identity can access the private information stored therein.
4

5 **BRIEF DESCRIPTION OF THE DRAWINGS**

6 A more complete understanding of exemplary methods and arrangements
7 of the present invention may be had by reference to the following detailed
8 description when taken in conjunction with the accompanying drawings wherein:

9 Fig. 1 is a diagram of an exemplary computer system on which the
10 described embodiments may be implemented.

11 Fig. 2 is a flow diagram of a method according to a general implementation
12 in which an Internet context is changed from an Internet context for a first identity
13 to an Internet context for a second identity.

14 Fig. 3 is a block diagram of a computer having memory storage.

15 Fig. 4 is a depiction of an index file having multiple records.

16 Fig. 5 is a simplified diagram of a registry in the system of Fig. 3.

17 Fig. 6 is a flow diagram illustrating a method for handling shared and user-
18 specific Internet content.
19
20
21
22
23
24
25

1 **DETAILED DESCRIPTION**

2 The invention is illustrated in the drawings as being implemented in a
3 suitable computing environment. Although not required, the invention will be
4 described in the general context of computer-executable instructions, such as
5 program modules, to be executed by a computing device, such as a personal
6 computer or a hand-held computer or electronic device. Generally, program
7 modules include routines, programs, objects, components, data structures, etc. that
8 perform particular tasks or implement particular abstract data types. Moreover,
9 those skilled in the art will appreciate that the invention may be practiced with
10 other computer system configurations, including multi-processor systems,
11 microprocessor-based or programmable consumer electronics, network PCs,
12 minicomputers, mainframe computers, and the like. The invention may also be
13 practiced in distributed computing environments where tasks are performed by
14 remote processing devices that are linked through a communications network. In
15 a distributed computing environment, program modules may be located in both
16 local and remote memory storage devices.

17
18 **Exemplary Computer Environment**

19 The various components and functionality described herein are
20 implemented with a number of individual computers. Fig. 1 shows components of
21 typical example of such a computer, referred by to reference numeral 100. The
22 components shown in Fig. 1 are only examples, and are not intended to suggest
23 any limitation as to the scope of the functionality of the invention; the invention is
24 not necessarily dependent on the features shown in Fig. 1.

Generally, various different general purpose or special purpose computing system configurations can be used. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

The functionality of the computers is embodied in many cases by computer-executable instructions, such as program modules, that are executed by the computers. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Tasks might also be performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media.

The instructions and/or program modules are stored at different times in the various computer-readable media that are either part of the computer or that can be read by the computer. Programs are typically distributed, for example, on floppy disks, CD-ROMs, DVD, or some form of communication media such as a modulated signal. From there, they are installed or loaded into the secondary memory of a computer. At execution, they are loaded at least partially into the computer's primary electronic memory. The invention described herein includes these and other various types of computer-readable media when such media contain instructions programs, and/or modules for implementing the steps

information such as computer-readable instructions, data structures, program modules, or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer 110. Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term “modulated data signal” means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer 100, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of example, and not limitation, Fig. 1 illustrates operating system 134, application programs 135, other program modules 136, and program data 137.

1 The computer 100 may also include other removable/non-removable,
2 volatile/nonvolatile computer storage media. By way of example only, Fig. 1
3 illustrates a hard disk drive 141 that reads from or writes to non-removable,
4 nonvolatile magnetic media, a magnetic disk drive 151 that reads from or writes to
5 a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that
6 reads from or writes to a removable, nonvolatile optical disk 156 such as a CD
7 ROM or other optical media. Other removable/non-removable,
8 volatile/nonvolatile computer storage media that can be used in the exemplary
9 operating environment include, but are not limited to, magnetic tape cassettes,
10 flash memory cards, digital versatile disks, digital video tape, solid state RAM,
11 solid state ROM, and the like. The hard disk drive 141 is typically connected to
12 the system bus 121 through an non-removable memory interface such as interface
13 140, and magnetic disk drive 151 and optical disk drive 155 are typically
14 connected to the system bus 121 by a removable memory interface such as
15 interface 150.

16 The drives and their associated computer storage media discussed above
17 and illustrated in Fig. 1 provide storage of computer-readable instructions, data
18 structures, program modules, and other data for computer 100. In Fig. 1, for
19 example, hard disk drive 141 is illustrated as storing operating system 144,
20 application programs 145, other program modules 146, and program data 147.
21 Note that these components can either be the same as or different from operating
22 system 134, application programs 135, other program modules 136, and program
23 data 137. Operating system 144, application programs 145, other program
24 modules 146, and program data 147 are given different numbers here to illustrate
25 that, at a minimum, they are different copies. A user may enter commands and

1 information into the computer 100 through input devices such as a keyboard 162
2 and pointing device 161, commonly referred to as a mouse, trackball, or touch
3 pad. Other input devices (not shown) may include a microphone, joystick, game
4 pad, satellite dish, scanner, or the like. These and other input devices are often
5 connected to the processing unit 120 through a user input interface 160 that is
6 coupled to the system bus, but may be connected by other interface and bus
7 structures, such as a parallel port, game port, or a universal serial bus (USB). A
8 monitor 191 or other type of display device is also connected to the system bus
9 121 via an interface, such as a video interface 190. In addition to the monitor,
10 computers may also include other peripheral output devices such as speakers 197
11 and printer 196, which may be connected through an output peripheral interface
12 195.

13 The computer may operate in a networked environment using logical
14 connections to one or more remote computers, such as a remote computer 180.
15 The remote computer 180 may be a personal computer, a server, a router, a
16 network PC, a peer device or other common network node, and typically includes
17 many or all of the elements described above relative to computer 100, although
18 only a memory storage device 181 has been illustrated in Fig. 1. The logical
19 connections depicted in Fig. 1 include a local area network (LAN) 171 and a wide
20 area network (WAN) 173, but may also include other networks. Such networking
21 environments are commonplace in offices, enterprise-wide computer networks,
22 intranets, and the Internet.

23 When used in a LAN networking environment, the computer 100 is
24 connected to the LAN 171 through a network interface or adapter 170. When used
25 in a WAN networking environment, the computer 100 typically includes a modem

1 172 or other means for establishing communications over the WAN 173, such as
2 the Internet. The modem 172, which may be internal or external, may be
3 connected to the system bus 121 via the user input interface 160, or other
4 appropriate mechanism. In a networked environment, program modules depicted
5 relative to the computer 100, or portions thereof, may be stored in the remote
6 memory storage device. By way of example, and not limitation, Fig. 1 illustrates
7 remote application programs 185 as residing on memory device 181. It will be
8 appreciated that the network connections shown are exemplary and other means of
9 establishing a communications link between the computers may be used.

10 Fig. 2 is a flow diagram of a method according to a general implementation
11 in which an Internet context is changed from an Internet context for a first identity
12 to an Internet context for a second identity. At step 200, a client computer
13 operates within a context for a first identity. An Internet context comprises data
14 associated with a particular user, or identity. Such data includes a history of web
15 sites that a particular user has visited, cookies that have been placed on the
16 computer by a particular user that allows a web site to gather information about
17 that user, web page content, etc.

18 At step 202, a request is received to switch from the first identity to a
19 second identity. The request is received by an Internet management object
20 operating on the client computer. The Internet management object is responsible
21 for maintaining an Internet connection and managing the data associated
22 therewith. Although described herein as an Internet management object, an object
23 may be more generally described as a network management object, which
24 performs similar task on networks other than the Internet, such as an intranet, a
25 local area network, a wide area network, etc.

identity 318. The current identity 318 contains a pointer to one of the identities 312-316 that is the current identity. Each identity includes an identity name, an optional identity password, and an identity identifier. The identity identifier is stored as a globally unique identifier (guid) in accordance with a conventional identification scheme in which only one name is associated with a particular object. Each identity also typically includes other work-environment information that can be used by identity-aware applications.

Wininet 310 contains pointers to containers that make up an Internet context. Wininet 310 includes a cookies pointer 320, a history pointer 322, a TIF (temporary Internet files) pointer 324, and a user-defined pointer 326. It is noted that there may be more than one user-defined container and pointer thereto. However, for discussion purposes, only one user-defined pointer is shown herein.

Wininet 310 also includes parameters associated with Internet browsing, such as a session start time 328 and authentication credentials 330. It is noted that these parameters are exemplary only and there may be other or additional parameters, depending on the implementation.

The Wininet pointers reference memory locations in the hierarchical directory structure 306. The cookies pointer 320 references a cookies directory 332; the history pointer 322 references a history directory 334; the TIF pointer 324 references a TIF directory 336; and the user-defined pointer 326 references a user-defined directory 338. It is noted that the cookies directory 332, the history directory 334, the TIF directory 336 and the user-defined directory 338 are subdirectories of at least one superior directory in the directory hierarchy. However, superior directories in the structure are not shown for convenience purposes.

The cookies directory 332, the history directory 334 and the TIF directory 336 are default containers that are associated with identity 0 312. To distinguish Internet context for one identity from Internet context for another identity, subdirectories are created that are associated with each directory referenced by Wininet 310 (except for the TIF directory 336, which will be explained in greater detail below).

The cookies directory 332 has two subdirectories, cookies\1 340 and cookies\2 342. Directory cookies\1 340 is associated with identity 1 314, and directory cookies\2 342 is associated with identity 2 316. The history directory 334 has two subdirectories, history\1 344 and history\2 346. The user-defined directory 338 has two subdirectories, user\1 348 and user\2 350. Each subdirectory is uniquely associated with an identity. It is noted that each directory 332-338 may have virtually any number of subdirectories, according to the number of identities on the system. However, for discussion purposes, the number of identities and, thus, the number of subdirectories is limited herein to two.

Wininet 310 receives requests to switch identities, e.g., to switch Internet contexts from identity 1 314 to identity 2 316. Each request includes a guid, or identifier, associated with identity 2 316. If the request includes an invalid guid, no switch occurs and the browsing continues with the current identity.

More particularly, Wininet 310 receives an InternetSetOption which includes the guid for identity 2 316 and a setting called INTERNET_OPTION_IDENTITY. The Internet context for identity 1 314 (except for content) is stored in containers associated with the guid for identity 1 314. Cookies associated with identity 1 314 are stored in the directory structure 306 in directory cookies\1 340, which is associated with identity 1 314. Historical

data associated with identity 1 314 is stored in the directory history\1 342. User-defined context data for identity 1 314 is stored in directory user\1 348. Data associated with identity 1 314 is flushed. This includes authentication credentials 330 for identity 1 314, session cookies, and the like. At this point, the session start time 328 is reset to zero for the new identity.

The discussion so far relates only to non-content context. Management of Internet content will be described in greater detail below.

The TIF directory 336 contains an index file 400. The index file 400 is shown in greater detail in Fig. 4. The index file 400 includes one or more records 402a-402n. The records 402 are similar in structure, and will be referred to in the singular as record 402. Record 402 includes several members, or fields. As shown for discussion purposes, record 402 contains a URL field 404, a file name field 406, and a date field 408. Other fields or members may be included depending on the implementation. The features of the index file 400 will be discussed in greater detail, below, in discussion of the operation of one or more described implementations.

Fig. 5 is a simplified diagram of one implementation of a registry 308. Registry 308 is a hierarchical database of user and system information. A basic unit of information in the registry is called a “key.” Within each “key” there are smaller units of information called “subkeys.” One of the keys, key 500 (the HKEY_USERS key) contains all the user profiles. The subkey DEFAULT 502 is the subkey for a default profile. In this example, the subkey PROFILE 1 504 is the subkey for a first profile and the subkey PROFILE 2 506 is the subkey for the second profile. Registry 306 also has another key, key 508 (the HKEY_CURRENT_USER key), the value of which points to the one of the

1 HKEY_USERS subkeys (profiles) that is currently in use.
2 HKEY_CURRENT_USER 508 contains numerous subkeys 510, one of which is
3 the IDENTITIES subkey 512. The IDENTITIES subkey 512 contains a
4 CURRENT_IDENTITY 514 and an individual subkey 516 and 518 for each
5 configured identity. In this example, IDENTITY 1 is identity 1 314 in Fig. 3 and
6 IDENTITY 2 is identity 2 316 in Fig. 3. CURRENT_IDENTITY 514 is a value
7 that points to the one of these two identities 516, 518 that is currently in use. If no
8 identity is selected for use, then the system is in a "non-identity state" and
9 CURRENT-IDENTITY subkey 514 points to no identity.

10 There is also a TIF subkey 520, a COOKIES subkey 522, a HISTORY
11 subkey 524 and a USER-DEFINED subkey 526. The TIF subkey 520 contains a
12 value that points to a location of Internet content, or temporary internet files, for
13 the CURRENT_IDENTITY 514. The COOKIES subkey 522 contains a value that
14 points to a location of cookies for the CURRENT_IDENTITY 514. The
15 HISTORY subkey 524 contains a value that points to a location of history data for
16 the CURRENT_IDENTITY 514. The USER-DEFINED subkey 526 contains a
17 value that points to a location of user-defined data for the CURRENT_IDENTITY
18 514. It is noted that there may be more than one USER-DEFINED subkey, or
19 there may be no USER-DEFINED subkey depending on the implementation. For
20 the present discussion, only one USER-DEFINED subkey is shown.

21 When a switch request is submitted, it is determined if the identity is a new
22 identity or if it has been used before. If the identity has not been used before, then
23 the registry subkey values for COOKIES 522, HISTORY 524 and USER-
24 DEFINED 526 are set to new values and an Internet context begins to be built for
25 the new user. An ordinal x is generated for the guid, and the new value for

The content is determined to be shared or user-specific at step 608. If the content is shared, i.e., not user-specific (“NO” branch, step 608), then the content is stored, the memory location is entered into the index at the entry determined by the hash, and the pointers are set to the memory locations (step 610). The user then continues browsing at step 618.

If the content is user-specific (“YES” branch, step 608), then ID_CACHE_ENTRY of the index record determined by the hash is set at step 612. This indicates that the Internet content associated with this record is user-specific. At step 614, a combination of the URL and the ordinal associated with the current identity is hashed to determine an index entry that indicates where the user-specific content will be stored. At step 616, the content is stored, the memory location is entered into the index at the entry determined by the hash, and the pointers are set to the memory locations.

If the Internet content is already stored in the cache (“YES” branch, step 602, then the URL of the content is hashed at step 620 to determine an index entry that indicates the memory location in which the content can be found. The ID_CACHE_ENTRY field of the index record identified by the hash is checked at step 622 to determine if the content is shared or user-specific. If the content is shared, i.e., not user-specific (“NO” branch, step 622), then the pointers are set to the memory location indicated by the index entry located using the hash (step 626) and the browsing continues at step 618.

If the content is user-specific (“YES” branch, step 622), then a combination of the URL and the ordinal associated with the current identity is hashed at step 624 to determine an index entry that indicates the memory location of the user-

1 specific content. The pointers are then set to this memory location and the
2 browsing continues at step 618.

3 4 Conclusion

5 Utilizing the described implementations for managing Internet context,
6 switching Internet contexts can be accomplished without a current user having to
7 first log off and a second user log on. To a user, it simply appears as if the browser
8 has been restarted to begin a new session without having terminated the browser.
9 Internet data that is specific to particular users is managed so that the browser can
10 better identify an Internet context with a particular identity, or user.